Components

Schemas

AnimalSpeciesDTO

Properties

Name	Description	Schema
speciesName *required*		string

AnimalDTO

Properties

Name	Description	Schema
name *required*		string
age *required*		integer (int32)
animalSpecies *required*		string
description *required*		string
needsAttention *required*		boolean
needsOutdoorSpace *required*		boolean
socialAnimal *optional*		boolean
goodWithChildren *optional*		boolean

AdoptionDTO

Adoption Data Transfer Object

Properties

Name	Description	Schema
adopterName *required*	Adopter's name	string
animalName *required*	Animal's name	string

PutAdopterDto

DTO for putting adopter

Properties

Name	Description	Schema
phone *required*	Adopter's phone number	string
street *required*	Adopter's street	string
city *required*	Adopter's city	string
state *required*	Adopter's state	string
zip *required*	Adopter's zip	string
country *required*	Adopter's country	string

RoleRequestDTO

Details to change role

Properties

Name	Description	Schema
user *optional*	User email	string
role *optional*	Role name	string
operation *optional*	Role operation	enum (GRANT,REMOVE)

Adopter

Properties

Name	Description	Schema
id *optional*	*read only*	integer (int64)
phone *optional*		string
adoptions *optional*	*unique items*	< Adoption > array
user *optional*		User
street *optional*		string
city *optional*		string
state *optional*		string
zip *optional*		string

Name	Description	Schema
country *optional*		string

Adoption

Properties

Name	Description	Schema
id *optional*		integer (int64)
adoptionDate *optional*		string (date)
adopter *optional*		Adopter
animal *optional*		Animal

Animal

Properties

Name	Description	Schema
id *optional*		integer (int64)
needsAttention *optional*		boolean
needsOutdoorSpace *optional*		boolean
name *optional*		string
age *optional*		integer (int32)
description *optional*		string
animalSpecies *optional*		AnimalSpecies
adoptions *optional*	*unique items*	< Adoption > array
socialAnimal *optional*		boolean
goodWithChildren *optional*		boolean

AnimalSpecies

Properties

Name	Description	Schema
id *optional*		integer (int64)
speciesName *optional*		string
animals *optional*		< [Animal](#) > array

User

Details to create user

Properties

Name	Description	Schema
id *optional*	*read only*	integer (int64)
name *required*		string
lastname *required*		string
email *required*		string
password *required*	*write only*	string
roles *optional*		< string > array
adapter *optional*		[Adopter](#)

AuthenticationDTO

Details to authenticate

Properties

Name	Description	Schema
email *optional*	User email	string
password *optional*	User password	string

AuthenticationResponse

Properties

Name	Description	Schema
jwt *optional*		string

ChangePassDTO

New password

Properties

Name	Description	Schema
new_password *optional*	New Password	string

PasswordChanged

Properties

Name	Description	Schema
email *optional*		string
status *optional*		string

AdoptionFormDTO

Adoption Form Data Transfer Object

Properties

Name	Description	Schema
hasBigGarden *optional*	is the adopter's house big enough for the animal?	boolean
hasOtherPets *optional*	is the adopter have other pets?	boolean
hasMuchFreeTime *optional*	is the adopter have much free time?	boolean
hasChildren *optional*	is the adopter have children?	boolean
preferredAge *optional*	animal age for adoption	integer (int32)
preferredSpecies *optional*	animal name for adoption	string

AnimalSpeciesWithIdDTO

Properties

Name	Description	Schema
speciesName *required*		string
id *optional*		integer (int64)

GetAdopterDTO

DTO for getting adopter

Properties

Name	Description	Schema
id *optional*	Adopter ID	integer (int64)
email *optional*		string

GetAdopterByIdDTO

DTO for getting adopter by ID

Properties

Name	Description	Schema
id *optional*	Adopter ID	integer (int64)
name *optional*	Adopter name	string
lastname *optional*	Adopter lastname	string
email *optional*	Adopter email	string
phone *optional*	Adopter phone	string

OpenAPI definition

vv0 :revnumber: v0 :openapi: 3.0.1

Overview

Tags

Employee

Endpoints for employee management

Users

Endpoints for user management

Admin

Endpoints for admin management

Animal Species

Endpoints for managing animal species

Adopters

Endpoints for managing adopters

Animals

Endpoints for managing animals

Paths

GET `/api/animalspecies/{id}` Find animal species by ID

Returns an AnimalSpeciesDTO object

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
409	Species name already exists	No Links
404	Animal species not found	No Links
200	Successfully retrieved the animal species *Content* **application/json**	No Links

PUT `/api/animalspecies/{id}` Update animal species

Updates an existing AnimalSpeciesDTO object

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
409	Species name already exists	No Links
200	Successfully updated the animal species *Content* **application/json**	No Links

Code	Description	Links
404	Animal species not found	No Links

DELETE `/api/animalspecies/{id}` **Delete animal species**

Deletes an AnimalSpeciesDTO object by ID

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
409	Species name already exists	No Links
204	Successfully deleted the animal species	No Links
404	Animal species not found	No Links

GET `/api/animal/{id}`

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/api/animal/{id}` **Update animal**

Updates an existing AnimalDTO object

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	Successfully updated the animal *Content* **application/json**	No Links
404	Animal not found	No Links

DELETE `/api/animal/{id}`

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK	No Links

GET `/api/adoptions/{id}`

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/api/adoptions/{id}`

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/api/adoptions/{id}`

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK	No Links

PUT `/api/adopters/{userId}` Update adopter

Parameters

Type	Name	Description	Schema
path	**userId** *required*		integer (int64)

Responses

Code	Description	Links
200	Successfully updated adopter *Content* **application/json**	No Links
500	Internal server error	No Links
404	Adopter not found	No Links

PUT `/api/admin/user/role` Endpoint for changing user role

Responses

Code	Description	Links
200	User role changed *Content* /	No Links
400	User role not changed *Content* /	No Links

POST /api/auth/signup Register a new user

Responses

Code	Description	Links
400	User account not created *Content* /	No Links
200	User created *Content* /	No Links

POST /api/auth/signin Endpoint for authentication

Responses

Code	Description	Links
400	User not authenticated *Content* /	No Links
200	User authenticated *Content* /	No Links

POST /api/auth/changepass Endpoint for changing password

Responses

Code	Description	Links
400	Password not changed *Content* /	No Links
200	Password changed *Content* /	No Links

GET `/api/animalspecies` **Find all animal species**

Returns a list of AnimalSpeciesWithIdDTO objects

Parameters

Type	Name	Description	Schema
query	**filter** *optional*	Filter criteria in the format: property[.operation]:value. Supported operations: eq, neq, gt, lt. Example: speciesName.eq=dog	string
query	**size** *optional*	Number of elements per page	integer (int32)
query	**page** *optional*	Page number, starting from 0	integer (int32)
query	**sort** *optional*	Sorting criteria in the format: property(,asc	desc). Default sort order is ascending. Multiple sort criteria are supported.
string	**query**	**direction** *optional*	Sorting order in the format: asc

Responses

Code	Description	Links
409	Species name already exists	No Links
500	Internal server error	No Links
200	Successfully retrieved list of animal species *Content* **application/json**	No Links

POST `/api/animalspecies` Create a new animal species

Saves a new AnimalSpeciesDTO object

Responses

Code	Description	Links
409	Species name already exists	No Links
200	Successfully created the animal species *Content* **application/json**	No Links
400	Invalid input	No Links

GET `/api/animal`

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/api/animal`

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/api/adoptions`

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST /api/adoptions

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST /api/adoptions/matchAnimal **Match adopter to animal**

Returns a map where key is a string representation of AnimalDTO and value is the match rating as double.

Parameters

Type	Name	Description	Schema
query	**size** *optional*		integer (int32)
query	**page** *optional*		integer (int32)

Responses

Code	Description	Links
500	Internal server error	No Links
200	Successfully matched adopter to animal *Content* **application/json**	No Links

POST /api/adoptions/adopt/{animalId} **Adopt an animal**

Parameters

Type	Name	Description	Schema
query	**id animal** *required*		integer (int64)

Responses

Code	Description	Links
500	Internal server error	No Links

Code	Description	Links
201	Successfully adopted animal *Content* **application/json**	No Links
400	Animal not found or Animal already adopted or User not found or User is not an adopter.	No Links

GET /api/empl/protected Endpoint for authentication

Responses

Code	Description	Links
400	User not authenticated *Content* /	No Links
200	User authenticated *Content* /	No Links

GET /api/empl/authenticated Endpoint for authentication

Responses

Code	Description	Links
400	User not authenticated *Content* /	No Links
200	User authenticated *Content* /	No Links

GET /api/adopters Find all adopters

Parameters

Type	Name	Description	Schema
query	**filter** *optional*		string

Type	Name	Description	Schema
query	**size** *optional*		integer (int32)
query	**page** *optional*		integer (int32)
query	**sort** *optional*		string
query	**direction** *optional*		string

Responses

Code	Description	Links
500	Internal server error	No Links
200	Successfully retrieved list of adopters *Content* **application/json**	No Links

GET `/api/adopters/{id}` Find adopter by ID

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	Successfully retrieved adopter *Content* **application/json**	No Links
500	Internal server error	No Links
404	Adopter not found	No Links

GET `/api/admin/user` Endpoint for getting all users from db

Responses

Code	Description	Links
400	Users not retrieved *Content* /	No Links
200	Users retrieved *Content* /	No Links

GET `/api/admin/logs` **Endpoint for retrieving all saved logs from db**

Responses

Code	Description	Links
200	Logs retrieved *Content* /	No Links
400	Logs not retrieved *Content* /	No Links

DELETE `/api/admin/user/{email}` **Endpoint for deleting user from db**

Parameters

Type	Name	Description	Schema
path	**email** *required*	User email	string

Responses

Code	Description	Links
200	User deleted *Content* /	No Links

Code	Description	Links
400	User not deleted *Content* /	No Links

Code	Description	Links
400	User not deleted *Content*	No Links